
	<p>Python Manipulation de chaîne de caractères</p>	 Année 2025/2026
---	--	---

A. Création et accès aux chaînes de caractères

Création d'une chaîne de caractères :

```

chaîne1 = "Bonjour tout le monde"
chaîne2 = 'Python est cool'
chaîne_multi_ligne = """Ceci est
une chaîne
sur plusieurs lignes."""

```

Accès aux caractères (indexation) :

```

texte = "Bonjour ça va ?"
print(texte[0])
print(texte[4])
print(texte[6])
print(texte[11])
print(texte[-1])

```

Résultat :

```

B
o
u
r
ç
a
v
a
?

```

L'index '-1' permet d'afficher le dernier caractère de la chaîne de caractère.

Les index :

B	o	n	j	o	u	r		ç	a		v	a		?
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Tranchage (slicing) :

```
phrase = "Programmation"
print(phrase[0:7])    #Affiche les caractères d'index 0 à 6
print(phrase[4:])    #Affiche du 4ème caractère (index) à la fin
print(phrase[:5])    #Affiche du début au 4ème caractère (index)
print(phrase[::2])   #Affiche tous les 2 caractères
print(phrase[::-1])  #Affiche tout à l'envers
```

Résultat :

```
Program
rammation
Progr
Pormain
noitamargorP
```

.lower()

Cette fonction permet de convertir tous les caractères en minuscules.

```
chaine = "BONJOUR tout le monde !"
print(chaine.lower())
```

Résultat :

```
bonjour tout le monde !
```

.upper()

Cette fonction permet de convertir tous les caractères en majuscules.

```
chaine = "Salut tout le monde"
print(chaine.upper())
```

Résultat :

```
SALUT TOUT LE MONDE
```

capitalize()

Cette fonction permet de mettre la première lettre de la chaîne en majuscule et le reste en minuscule.

```
chaîne = "python est TROP bien"
```

```
print(chaîne.capitalize())
```

Résultat :

```
Python est trop bien
```

.title()

Cette fonction permet de mettre la première lettre de chaque mot en majuscule.

```
chaîne = "manger des fruits et des légumes"
```

```
print(chaîne.title())
```

Résultat :

```
Manger Des Fruits Et Des Légumes
```

swapcase()

Cette fonction permet d'inverser la casse de tous les caractères. Les minuscules deviennent majuscules et les majuscules deviennent minuscules.

```
chaîne = "PyThOn"
```

```
print(chaîne.swapcase())
```

Résultat :

```
pYtHoN
```

Longueur d'une chaîne de caractère

La fonction len() permet de retourner la longueur de la chaîne de caractère mise en paramètre.

```
chaîne = "programmation"
```

```
longueur = len(chaîne)
```

```
print(longueur)
```

Résultat :

```
13
```

Compter le nombre d'occurrence

La fonction count() permet de compter les occurrences exactes d'une sous-chaîne de caractère.

```
chaine = "le chemin le plus long n'est pas le plus court"
nombre_le = chaine.count("le")
print(nombre_le)
```

Résultat :

3

Dans cet exemple, on compte le nombre de fois où apparaît le mot "le"

Recherche de chaîne de caractère

La fonction find() permet de rechercher la position (index) d'une chaîne de caractère. La fonction renvoie l'index d'où se trouve le terme, si le terme ne s'y trouve pas la fonction renvoie -1.

```
chaine = "BAC PRO CIEL à Loches"
mot_ciel = chaine.find("CIEL")
mot_banane = chaine.find("banane")
print(mot_ciel)
print(mot_banane)
```

Résultat :

8

-1

B	A	C		P	R	O		C	I	E	L		à		L	O	C	H	E	S
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Le mot banane n'apparaît pas dans la chaîne d'où le résultat « -1 »

Si on veut seulement savoir si un mot existe, on peut aussi utiliser in :

```
if "CIEL" in chaine :
    print("Mot trouvé")
```

Tests sur chaînes de caractères

La fonction `isalpha()` permet de tester si tous les caractères sont alphabétiques.

La fonction `isdigit()` permet de tester si tous les caractères sont des chiffres.

La fonction `isalnum()` permet de tester si tous les caractères sont alphanumériques (lettres ou chiffres).

La fonction `isspace()` permet de tester si tous les caractères sont des espaces blancs.

```
print("Python".isalpha())      # True
print("P2th0n".isalpha())     # False

print("1235".isdigit())       # True
print("P1230".isdigit())     # False

print("Pyth012n".isalnum())   # True
print("P123@".isalnum())     # False

print(" ".isspace())         # True
print("le 2".isspace())      # False
```

Supprimer les espaces blancs

La fonction `strip()` permet de supprimer les espaces blancs (espaces, tabulations retours à la ligne) au début et à la fin de la chaîne (pas au milieu de la chaîne).

```
chaine_sale = "  Bonjour le monde  \n"
print(chaine_sale)
print(chaine_sale.strip())
```

Résultat :

```
Bonjour le monde
Bonjour le monde
```

Remplacement d'occurrence

La fonction `replace(old,new)` permet de remplacer toutes les occurrences d'une sous-chaîne par une nouvelle sous-chaîne.

```
phrase = "Les pommes sont des légumes. J'aime les légumes."  
nouvelle_phrase = phrase.replace("légumes", "fruits")  
print(nouvelle_phrase)
```

Résultat :

```
Les pommes sont des fruits. J'aime les fruits.
```

Division de chaîne

La fonction `split()` permet de diviser la chaîne en une liste de sous-chaînes en utilisant un séparateur donné. Si aucun séparateur n'est fourni, elle divise par les espaces blancs multiples.

```
phrase = "Ceci est une phrase"  
liste_mots = phrase.split(" ")  
print(liste_mots)  
  
csv_data = "Jean,23,Loches"  
liste_csv = csv_data.split(",")  
print(liste_csv)
```

Résultat :

```
['Ceci', 'est', 'une', 'phrase']  
['Jean', '23', 'Loches']
```

Jointure de chaîne depuis une liste

La méthode `join()` permet de joindre les éléments d'une liste en une seule chaîne en précisant le séparateur.

```
liste_dossier = ["etc" , "var" , "www" , "html"]  
chaine_construite = "/" .join(liste_dossier)  
print(chaine_construite)
```

```
liste_num = ["192" , "168" , "1" , "254" ]  
ip_construite = "." .join(liste_num)  
print(ip_construite)
```

Résultat

```
etc/var/www/html  
192.168.1.254
```

B. Erreurs fréquentes

Index hors limites :

```
texte = "Python"  
print(texte[20])
```

IndexError: string index out of range

Confondre index et position :

```
texte = "Python"  
print(texte[1])
```

Résultat :

y

Oublier les parenthèses d'une méthode :

```
chaine = "BONJOUR"  
print(chaine.lower)
```

Résultat : affichage de l'objet méthode, pas du texte transformé

C. A retenir

une chaîne est une suite de caractères

le premier caractère est à l'index 0

un index négatif permet de partir de la fin

le slicing permet d'extraire une partie de la chaîne

lower(), upper(), capitalize(), title() modifient la casse

len() donne la longueur

split() découpe une chaîne

join() reconstruit une chaîne à partir d'une liste

D. Synthèse

